

Home

Java

Servlet

JSP

Struts2

Hibernate

Spring

MyBatis

Java WS

C

C++

C#

Python

PHP

Excel

VBA

Web

SQL

Misc

Python cơ bản

Câu Lệnh Điều Khiển

Python Tkinter(GUI)

Python Web Blocker

Python Numpy

Python Django

Python Flask

Bài tập Python

Bài tập Python có lời giải

Bài tập quản lý sinh viên trong Python

Bài Tập Python Kinh Điển

Cao

Bài tập quản lý sinh viên trong Python

Bài tập Python có lời giải

Chuỗi (String) trong Python

Giải phương trình bậc 2 trong Python

Bài này cung cấp cho bạn danh sách các dạng bài tập khác nhau để bạn thực hành khi học lập trình Python. Các code mẫu của **bài tập Python có lời giải** được viết trên Python 3.

Nội dung chính

Các cấp độ bài tập Python

1. Bài tập Python kinh điển

2. Bài tập Python cơ bản

3. Bài tập vòng lặp trong Python

Các cấp độ bài tập Python

Bài này cung cấp cho bạn danh sách các **bài tập Python có lời giải ở các cấp độ** khác nhau để bạn thực hành khi học ngôn ngữ lập trình C:

1. Bài tập Python kinh điển.

2. Bài tập Python cơ bản.

3. Bài tập vòng lặp trong Python.

4. Bài tập Python về mảng một chiều và mảng 2 chiều (ma trận).

5. Bài tập chuỗi trong Python.

6. Bài tập Python về đệ quy.

7. Bài tập Python về các thuật toán sắp xếp.

8. Bài tập Python về đọc ghi file.

1. Bài tập Python kinh điển

1. Dãy số Fibonacci trong Python

2. Check Số nguyên tố trong Python

3. Tính giai thừa trong Python

4. Chuyển đổi hệ cơ số trong Python

Bài 01: Viết một chương trình Python in ra dãy số Fibonacci

Gợi ý:

Sử dụng đệ quy

Không sử dụng đệ quy

Code mẫu: In dãy số Fibonacci trong Python không sử dụng đệ quy.

1"""

2\* Tính số fibonacci thứ n

3\*

4\* @param n: chỉ số của số fibonacci tính từ 0

Recent Updates

Về tam giác Pascal trong Python

Về tam giác Pascal trong Python

Về tam giác Pascal trong Python

Bài tập vòng lặp trong Python

Bài tập quản lý sinh viên trong Python

Bài tập Python có lời giải

Về tam giác Pascal trong Python

Check số nguyên tố trong Python

Cách cài đặt Python trên Windows

Hướng dẫn lập trình Python

Hướng dẫn lập trình Python

Community Edition

Hướng dẫn lập trình Python

Studio Code

VietTuts on

Học

4.379 likes

Thích Trang

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

https://viettuts.vn/bai-tap-python

1/11

```

5      *          vd: F0 = 0, F1 = 1, F2 = 1, F3 = 2
6      * @return số fibonacci thứ n
7      """
8      def fibonacci(n):
9          f0 = 0;
10         f1 = 1;
11         fn = 1;
12
13         if (n < 0):
14             return -1;
15         elif (n == 0 or n == 1):
16             return n;
17         else:
18             for i in range(2, n):
19                 f0 = f1;
20                 f1 = fn;
21                 fn = f0 + f1;
22             return fn;
23
24     print("10 số đầu tiên của dãy số Fibonacci: ");
25     sb = "";
26     for i in range(0, 10):
27         sb = sb + str(fibonacci(i)) + ", ";
28     print(sb)

```

Kết quả:

```

10 số đầu tiên của dãy số Fibonacci:
0 1 1 2 3 5 8 13 21 34

```

**Bài 02:** Viết một chương trình Python kiểm tra số nguyên tố.

Code mẫu:

```

1      """
2      * check số nguyên tố trong Python
3      *
4      * @author viettuts.vn
5      * @param n: số nguyên dương
6      * @return 1 là số nguyên tố,
7      *          0 không là số nguyên tố
8      """

```

Kết quả:

```

Các số nguyên tố nhỏ hơn 100 là:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

```

**Bài 03:** Viết một chương trình Python tính giai thừa của một số bằng cách không sử dụng đệ quy và có sử dụng đệ quy.

Gợi ý:

- Sử dụng đệ quy
- Không sử dụng đệ quy

Code mẫu: Tính giai thừa trong Python không sử dụng đệ quy.

```

1      """
2      * tính giai thừa
3      *
4      * @author viettuts.vn
5      * @param n: số nguyên dương
6      * @return giai thừa của số n
7      """
8      def tinhgiaithua(n):
9          giai_thua = 1;
10         if (n == 0 or n == 1):
11             return giai_thua;
12         else:
13             for i in range(2, n + 1):
14                 giai_thua = giai_thua * i;
15             return giai_thua;

```

```

16
17 n = int(input("Nhập số nguyên dương n = "));
18 print("Giai thừa của", n, "là", tinhgiaithua(n));

```

Kết quả:

```

Nhập số nguyên dương n = 10
Giai thừa của 10 là 3628800

```

**Bài 04:** Viết một chương trình Python để chuyển đổi số nguyên N sang hệ cơ số B ( $2 \leq B \leq 32$ ) bất kỳ.

Code mẫu: Chuyển đổi hệ cơ số 10 sang hệ cơ số B

```

1  """
2  # Chuyển đổi số nguyên n sang hệ cơ số b
3  #
4  # @author viettuts.vn
5  # @param n: số nguyên
6  # @param b: hệ cơ số
7  # @return hệ cơ số b
8  """
9  def convert_number(n, b):
10     if (n < 0 or b < 2 or b > 16):
11         return ""
12
13     sb = "";
14     m = 0;
15     remainder = n;
16
17     while (remainder > 0):
18         if (b > 10):
19             m = remainder % b;
20             if (m >= 10):
21                 sb = sb + str(chr(55 + m));
22             else:
23                 sb = sb + str(m);
24         else:
25             sb = sb + str(remainder % b);
26             remainder = int(remainder / b);
27     return "".join(reversed(sb)); # đảo ngược chuỗi sb
28
29 n = int(input("Nhập số nguyên dương n = "));
30 print("Hệ cơ số 2 của số nguyên ", n, "là:", convert_number(n, 2))
31 print("Hệ cơ số 16 của số nguyên ", n, "là:", convert_number(n, 16))

```

Kết quả:

```

Nhập số nguyên dương n = 15
Hệ cơ số 2 của số nguyên 15 là: 1111
Hệ cơ số 16 của số nguyên 15 là: F

```

## 2. Bài tập Python cơ bản

**Danh sách bài tập:**

- Viết chương trình tìm tất cả các số chia hết cho 7 nhưng không phải bội số của 5, nằm trong đoạn 10 và 200 (tính cả 10 và 200). Các số thu được sẽ được in thành chuỗi trên một dòng, cách nhau bằng dấu phẩy.
- Viết một chương trình tính giai thừa của một số nguyên dương n. Với n được nhập từ bàn phím. Ví dụ, n = 8 thì kết quả đầu ra phải là  $1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 = 40320$ .
- Hãy viết chương trình để tạo ra một dictionary chứa (i, i<sup>i</sup>), trong đó i là số nguyên từ 1 đến n (bao gồm cả 1 và n), n được nhập từ bàn phím. Sau đó in ra dictionary này ra màn hình. Ví dụ: Giả sử số n là 8 thì đầu ra sẽ là: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}.
- Viết chương trình Python giải phương trình bậc 2:  $ax^2 + bx + c = 0$ .
- Viết chương trình tìm ước số chung lớn nhất (USCLN) và bội số chung nhỏ nhất (BSCNN) của hai số nguyên dương a và b nhập từ bàn phím.
- Viết chương trình liệt kê tất cả các số nguyên tố nhỏ hơn n. Số nguyên dương n được nhập từ bàn phím.
- Viết chương trình liệt kê n số nguyên tố đầu tiên trong Python. Số nguyên dương n được nhập từ bàn phím.
- Viết chương trình liệt kê tất cả số nguyên tố có 5 chữ số trong Python.

9. Viết chương trình phân tích số nguyên n thành các thừa số nguyên tố trong Python. Ví dụ:  $100 = 2 \times 2 \times 5 \times 5$ .
10. Viết chương trình tính tổng của các chữ số của một số nguyên n trong Python. Số nguyên dương n được nhập từ bàn phím. Với  $n = 1234$ , tổng các chữ số:  $1 + 2 + 3 + 4 = 10$
11. Viết chương trình kiểm tra một số n là số thuận nghịch trong Python. Số nguyên dương n được nhập từ bàn phím.
12. Viết chương trình liệt kê các số Fibonacci nhỏ hơn n là số nguyên tố trong Python. N là số nguyên dương được nhập từ bàn phím.
13. Viết chương trình chấp nhận một chuỗi số, phân tách bằng dấu phẩy từ giao diện điều khiển, tạo ra một danh sách và một tuple chứa mọi số.

**Bài 01:**

Viết chương trình tìm tất cả các số chia hết cho 7 nhưng không phải bội số của 5, nằm trong đoạn 10 và 200 (tính cả 10 và 200). Các số thu được sẽ được in thành **chuỗi** trên một dòng, cách nhau bằng dấu phẩy.

Gợi ý:

- Sử dụng range(begin, end)

Code mẫu:

```
1 arr = []
2 for i in range(10, 201):
3     if (i % 7 == 0) and (i % 5 != 0):
4         arr.append(str(i))
5 print(','.join(arr))
```

Kết quả:

```
14, 21, 28, 42, 49, 56, 63, 77, 84, 91, 98, 112, 119, 126, 133, 147, 154, 161, 168,
182, 189, 196
```

**Bài 02:**

Viết một chương trình tính giai thừa của một số nguyên dương n. Với n được nhập từ bàn phím. Ví dụ,  $n = 8$  thì kết quả đầu ra phải là  $1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8 = 40320$ .

Gợi ý:

- Sử dụng đệ quy hoặc vòng lặp để tính giai thừa.

Code mẫu: sử dụng đệ quy

```
1 n = int(input("Nhập số cần tính giai thừa: "))
2
3 def giaiThua(n):
4     if n == 0:
5         return 1
6     return n * giaiThua(n - 1)
7
8 print(giaiThua(n))
```

Kết quả:

```
Nhập số cần tính giai thừa: 8
40320
```

**Bài 03:**

Hãy viết chương trình để tạo ra một **dictionary** chứa (i, i<sup>i</sup>), trong đó i là số nguyên từ 1 đến n (bao gồm cả 1 và n), n được nhập từ bàn phím. Sau đó in ra dictionary này ra màn hình. Ví dụ: Giả sử số n là 8 thì đầu ra sẽ là: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}.

Gợi ý:

- Sử dụng vòng lặp for để lặp i từ 1 đến n.

Code mẫu:

```
1 n = int(input("Nhập số nguyên dương n = "))
2
3 d = dict()
4 for i in range(1, n + 1):
5     d[i] = i * i
6
7 print (d)
```

Kết quả:

```
Nhập số nguyên dương n = 8
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}
```

#### Bài 04:

Viết chương trình giải phương trình bậc 2:  $ax^2 + bx + c = 0$ .

Code mẫu:

```
1 import math
2
3 """
4 # Giải phương trình bậc 2:  $ax^2 + bx + c = 0$ 
5 # @param a: hệ số bậc 2
6 # @param b: hệ số bậc 1
7 # @param c: số hạng tự do
8 """
9 def giaiPTBac2(a, b, c):
10     # kiểm tra các hệ số
11     if (a == 0):
12         if (b == 0):
13             print("Phương trình vô nghiệm!");
14         else:
15             print("Phương trình có một nghiệm: x = ", + (-c / b));
16             return;
17
18     # tính delta
19     delta = b * b - 4 * a * c;
20     # tính nghiệm
21     if (delta > 0):
22         x1 = (float)((-b + math.sqrt(delta)) / (2 * a));
23         x2 = (float)((-b - math.sqrt(delta)) / (2 * a));
24         print("Phương trình có 2 nghiệm là: x1 = ", x1, " và x2 = ", x2);
25     elif (delta == 0):
26         x1 = (-b / (2 * a));
27         print("Phương trình có nghiệm kép: x1 = x2 = ", x1);
28     else:
29         print("Phương trình vô nghiệm!");
30
31 # Nhập các hệ số
32 a = float(input("Nhập hệ số bậc 2, a = "));
33 b = float(input("Nhập hệ số bậc 1, b = "));
34 c = float(input("Nhập hằng số tự do, Python = "));
35 # Gọi hàm giải phương trình bậc 2
36 giaiPTBac2(a, b, c)
```

Kết quả:

```
Nhập hệ số bậc 2, a = 2
Nhập hệ số bậc 1, b = 1
Nhập hằng số tự do, Python = -1
Phương trình có 2 nghiệm là: x1 = 0.5 và x2 = -1.0
```

#### Bài 05:

Viết chương trình tìm ước số chung lớn nhất (USCLN) và bội số chung nhỏ nhất (BSCNN) của hai số nguyên dương a và b nhập từ bàn phím.

Gợi ý:

- Sử dụng giải thuật Euclid.

Code mẫu:

```

1  """
2  * Tìm ước số chung lớn nhất (USCLN)
3  *
4  * @param a: số nguyên dương
5  * @param b: số nguyên dương
6  * @return USCLN của a và b
7  """
8  def uscln(a, b):
9      if (b == 0):
10         return a;
11     return uscln(b, a % b);
12
13     """
14     * Tìm bội số chung nhỏ nhất (BSCNN)
15     *
16     * @param a: số nguyên dương
17     * @param b: số nguyên dương
18     * @return BSCNN của a và b
19     """
20     def bscnn(a, b):
21         return int((a * b) / uscln(a, b));
22
23     a = int(input("Nhập số nguyên dương a = "));
24     b = int(input("Nhập số nguyên dương b = "));
25     #tính USCLN của a và b
26     print("Ước số chung lớn nhất của", a, "và", b, "là:", uscln(a, b));
27     #tính BSCNN của a và b
28     print("Bội số chung nhỏ nhất của", a, "và", b, "là:", bscnn(a, b));

```

Kết quả:

```

Nhập số nguyên dương a = 15
Nhập số nguyên dương b = 20
Ước số chung lớn nhất của 15 và 20 là: 5
Bội số chung nhỏ nhất của 15 và 20 là: 60

```

#### Bài 06:

Viết chương trình liệt kê tất cả các số nguyên tố nhỏ hơn n. Số nguyên dương n được nhập từ bàn phím.

Code mẫu:

```

1  import math
2
3  """
4  * check so nguyen to
5  *
6  * @author viettuts.vn
7  * @param n: so nguyen duong
8  * @return true la so nguyen so,
9  *         false khong la so nguyen to
10 """
11 def isPrimeNumber(n):
12     # so nguyen n < 2 khong phai la so nguyen to
13     if (n < 2):
14         return False;
15
16     # check so nguyen to khi n >= 2
17     squareRoot = int(math.sqrt(n));
18     for i in range(2, squareRoot + 1):
19         if (n % i == 0):
20             return False;
21     return True;
22
23 n = int(input("Nhập số nguyên dương n = "));
24 print ("Tất cả các số nguyên tố nhỏ hơn", n, "là:");
25 sb = "";
26 if (n >= 2):
27     sb = sb + "2" + " ";
28     for i in range (3, n+1):
29         if (isPrimeNumber(i)):
30             sb = sb + str(i) + " ";

```

```

31 |         i = i + 2;
32 |     print(sb);

```

Kết quả:

Nhập số nguyên dương n = 100  
 Tất cả các số nguyên tố nhỏ hơn 100 là:  
 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

#### Bài 07:

Viết chương trình liệt kê n số nguyên tố đầu tiên trong Python. Số nguyên dương n được nhập từ bàn phím.

Code mẫu:

```

1 | import math
2 |
3 | """
4 | * check so nguyen to
5 | *
6 | * @author viettuts.vn
7 | * @param n: so nguyen duong
8 | * @return true la so nguyen so,
9 | *         false khong la so nguyen to
10 | """
11 | def isPrimeNumber(n):
12 |     # so nguyen n < 2 khong phai la so nguyen to
13 |     if (n < 2):
14 |         return False;
15 |
16 |     # check so nguyen to khi n >= 2
17 |     squareRoot = int(math.sqrt(n));
18 |     for i in range(2, squareRoot + 1):
19 |         if (n % i == 0):
20 |             return False;
21 |     return True;
22 |
23 | n = int(input("Nhập số nguyên dương n = "));
24 | print(n, "Số nguyên tố đầu tiên là:");
25 | dem = 0; # đếm số số nguyên tố
26 | i = 2;   # tìm số nguyên tố bắt đầu từ số 2
27 | sb = "";
28 | while (dem < n):
29 |     if (isPrimeNumber(i)):
30 |         sb = sb + str(i) + " ";
31 |         dem = dem + 1;
32 |         i = i + 1;
33 | print(sb);

```

Kết quả:

Nhập số nguyên dương n = 10  
 10 Số nguyên tố đầu tiên là:  
 2 3 5 7 11 13 17 19 23 29

#### Bài 8:

Viết chương trình liệt kê tất cả số nguyên tố có 5 chữ số trong Python.

Code mẫu:

```

1 | import math
2 |
3 | """
4 | * check so nguyen to
5 | *
6 | * @author viettuts.vn
7 | * @param n: so nguyen duong
8 | * @return true la so nguyen so,
9 | *         false khong la so nguyen to
10 | """
11 | def isPrimeNumber(n):

```

```

12     # so nguyen n < 2 khong phai la so nguyen to
13     if (n < 2):
14         return False;
15
16     # check so nguyen to khi n >= 2
17     squareRoot = int(math.sqrt(n));
18     for i in range(2, squareRoot + 1):
19         if (n % i == 0):
20             return False;
21     return True;
22
23     print ("Liệt kê tất cả số nguyên tố có 5 chữ số:");
24     dem = 0;
25     for i in range(10001, 99999):
26         if (isPrimeNumber(i)):
27             print(i);
28             dem = dem + 1;
29     print("Tổng các số nguyên tố có 5 chữ số là:", dem);

```

Kết quả:

```

Liệt kê tất cả số nguyên tố có 5 chữ số:
10007
10009
10037
...
99971
99989
99991
Tổng các số nguyên tố có 5 chữ số là: 8363

```

#### Bài 9:

Viết chương trình phân tích số nguyên n thành các thừa số nguyên tố trong Python. Ví dụ:  $100 = 2 \times 2 \times 5 \times 5$ .

Code mẫu:

```

1  """
2  * Phân tích số nguyên thành tích các thừa số nguyên tố
3  *
4  * @param positiveInt
5  * @return
6  """
7  def phanTichSoNguyen(n):
8      i = 2;
9      listNumbers = [];
10     # phân tích
11     while (n > 1):
12         if (n % i == 0):
13             n = int(n / i);
14             listNumbers.append(i);
15         else:
16             i = i + 1;
17     # nếu listNumbers trống thì add n vào listNumbers
18     if (len(listNumbers) == 0):
19         listNumbers.append(n);
20     return listNumbers;
21
22     n = int(input("Nhập số nguyên dương n = "));
23     # phân tích số nguyên dương n
24     listNumbers = phanTichSoNguyen(n);
25     size = len(listNumbers);
26     sb = "";
27     for i in range(0, size - 1):
28         sb = sb + str(listNumbers[i]) + " x ";
29     sb = sb + str(listNumbers[size-1]);
30     # in kết quả ra màn hình
31     print("Kết quả:", n, "=", sb);

```

Kết quả:

```

Nhập số nguyên dương n = 100
Kết quả: 100 = 2 x 2 x 5 x 5

```

#### Bài 10:



Viết chương trình tính tổng của các chữ số của một số nguyên n trong Python. Số nguyên dương n được nhập từ bàn phím. Với n = 1234, tổng các chữ số:  $1 + 2 + 3 + 4 = 10$

Code mẫu:

```
1  """
2  * Tính tổng của các chữ số của một số nguyên dương n
3  *
4  * @param n: số nguyên dương
5  * @return
6  """
7  def totalDigitsOfNumber(n):
8      total = 0;
9      while (n > 0):
10         total = total + n % 10;
11         n = int(n / 10);
12     return total;
13
14 n = int(input("Nhập số nguyên dương n = "));
15 print("Tổng các chữ số của", n , "là", totalDigitsOfNumber(n));
```

Kết quả:

```
Nhập số nguyên dương n = 1234
Tổng các chữ số của 1234 là 10
```

#### Bài 11:

Viết chương trình kiểm tra một số n là số thuận nghịch trong Python. Số nguyên dương n được nhập từ bàn phím.

Code mẫu:

```
1  """
2  * Kiểm tra số thuận nghịch
3  *
4  * @param n: số nguyên dương
5  * @return true là số thuận nghịch
6  *         false không là số thuận nghịch
7  """
8  def isThuanNghich(n):
9      str1 = str(n); # ep kieu so n thanh chuoai
10     str2 = str1[::-1]; # dao nguoc chuoai str1
11     if (str1 == str2):
12         return True;
13     else:
14         return False;
15
16 n = int(input("Nhập số nguyên dương n = "));
17 print("Tổng các chữ số của", n , "là", isThuanNghich(n));
18 m = int(input("Nhập số nguyên dương m = "));
19 print("Tổng các chữ số của", m , "là", isThuanNghich(m));
20 print("Tổng các chữ số của", n , "là", isThuanNghich(n));
```

Kết quả:

```
Nhập số nguyên dương n = 123321
Tổng các chữ số của 123321 là True
Nhập số nguyên dương m = 123451
Tổng các chữ số của 123451 là False
```

#### Bài 12:

Viết chương trình liệt kê các số Fibonacci nhỏ hơn n là số nguyên tố trong Python. N là số nguyên dương được nhập từ bàn phím.

Code mẫu:

```
1  import math
2
```

```

3  """
4  * Tính số fibonacci thứ n
5  *
6  * @param n: chỉ số của số fibonacci tính từ 0
7  *           vd: F0 = 0, F1 = 1, F2 = 1, F3 = 2
8  * @return số fibonacci thứ n
9  """
10 def fibonacci(n):
11     if (n < 0):
12         return -1;
13     elif (n == 0 or n == 1):
14         return n;
15     else:
16         return fibonacci(n - 1) + fibonacci(n - 2);
17
18
19 """
20 * check so nguyen to
21 *
22 * @author viettuts.vn
23 * @param n: so nguyen duong
24 * @return true la so nguyen so,
25 *         false khong la so nguyen to
26 """
27 def isPrimeNumber(n):
28     # so nguyen n < 2 khong phai la so nguyen to
29     if (n < 2):
30         return False;
31
32     # check so nguyen to khi n >= 2
33     squareRoot = int(math.sqrt(n));
34     for i in range(2, squareRoot + 1):
35         if (n % i == 0):
36             return False;
37     return True;
38
39
40 """
41 * Chương trình liệt kê các số Fibonacci nhỏ hơn n là số nguyên tố.
42 * Với n được nhập từ bàn phím.
43 *
44 * @author viettuts.vn
45 """
46 n = int(input("Nhập số nguyên dương n = "));
47 print ("Tất cả các số fibonacci nhỏ hơn", n, "và nguyên tố:");
48 i = 0;
49 fin = fibonacci(i);
50 while(fin < n):
51     fin = fibonacci(i);
52     if (isPrimeNumber(fin)):
53         print(fin)
54     i = i + 1;

```

Kết quả:

```

Nhập số nguyên dương n = 100
Tất cả các số fibonacci nhỏ hơn 100 và nguyên tố:
2
3
5
13
89

```

### Bài 13:

Viết chương trình chấp nhận một chuỗi số, phân tách bằng dấu phẩy từ giao diện điều khiển, tạo ra một danh sách và một tuple chứa mọi số.

Ví dụ: Đầu vào được cung cấp là 34, 67, 55, 33, 12, 98 thì đầu ra là:

```

['34', '67', '55', '33', '12', '98']
('34', '67', '55', '33', '12', '98')

```

Gợi ý:

- Viết lệnh yêu cầu nhập vào các giá trị sau đó dùng quy tắc chuyển đổi kiểu dữ liệu để hoàn tất.

Code mẫu:

```

1 values=input("Nhập vào các giá trị: ")
2 l=values.split(",")
3 t=tuple(l)
4 print (l)
5 print (t)

```

Kết quả:

```

Nhập vào các giá trị: 34, 67, 55, 33, 12, 98
['34', ' 67', ' 55', ' 33', ' 12', ' 98']
('34', ' 67', ' 55', ' 33', ' 12', ' 98')

```

### 3. Bài tập vòng lặp trong Python

Bài này tổng hợp các bài tập vòng lặp trong Python giúp bạn làm quen cú pháp và cách sử dụng của các vòng lặp Python cơ bản: **vòng lặp for** , **vòng lặp while** và vòng lặp lồng nhau.

- [Bài tập vòng lặp trong python](#)
- [Vẽ tam giác đều trong Python](#)
- [Vẽ tam giác vuông cân trong Python](#)
- [Vẽ tam giác Floyd trong Python](#)
- [Vẽ tam giác Pascal trong Python](#)

---

☺ Chuỗi (String) trong Python

Giải phương trình bậc 2 trong Python ☺

---

#### Danh sách bài học

---

[Học java](#)  
[Học servlet](#)  
[Học jsp](#)  
[Học Hibernate](#)  
[Học Struts2](#)  
[Học Spring](#)  
[Học SQL](#)

#### Câu hỏi phỏng vấn

---

[201 câu hỏi phỏng vấn java](#)  
[25 câu hỏi phỏng vấn servlet](#)  
[75 câu hỏi phỏng vấn jsp](#)  
[52 câu hỏi phỏng vấn Hibernate](#)  
[70 câu hỏi phỏng vấn Spring](#)  
[57 câu hỏi phỏng vấn SQL](#)

#### About VietTuts.Vn

---

Hệ thống bài học trên VietTuts.Vn bao gồm thuyết và thực hành về các công nghệ web. Các bài lý thuyết trên hệ thống V tham khảo và tổng hợp từ các trang hi <http://www.tutorialspoint.com>, <http://do>